Deterministic Gibbs Sampling for Data Association in Multi-Object Tracking

Laura M. Wolf and Marcus Baum

Abstract—In multi-object tracking, multiple objects generate multiple sensor measurements, which are used to estimate the objects’ state simultaneously. Since it is unknown from which object a measurement originates, a data association problem arises. Considering all possible associations is computationally infeasible for large numbers of objects and measurements. Hence, approximation methods are applied to compute the most relevant associations. Here, we focus on deterministic methods, since multi-object tracking is often applied in safety-critical areas. In this work we show that Herded Gibbs sampling, a deterministic version of Gibbs sampling, applied in the labeled multi-Bernoulli filter, yields results of the same quality as randomized Gibbs sampling while having comparable computational complexity. We conclude that it is a suitable deterministic alternative to randomized Gibbs sampling and could be a promising approach for other data association problems.

I. INTRODUCTION

The tracking of multiple moving objects based on measurements, Multi-Object Tracking (MOT), is a highly investigated field with safety relevance in application areas such as aerial and maritime surveillance as well as driver assistance systems and autonomous vehicles in the automotive area.

The measurements are obtained from miscellaneous sensors, e.g., radar, lidar or sonar sensors. Due to the respective sensor technology and environmental factors, the measurements are in general noise corrupted. Hence, existing objects are not always detected and false alarms can occur caused by clutter detection. Furthermore, in most scenarios the number of objects is unknown and dynamic, and it is unknown from which object a measurement originated (Fig. 1), since there could be multiple measurements close to one object (e.g., object 1 and measurements 1 and 2) or multiple objects close to the same measurement (e.g., objects 3 and 4 and measurement 4). These unknown quantities lead to a data association problem, which has exponential complexity [1].

In this work, we consider classical MOT, where each object is assumed to generate at most one measurement per scan. There exist numerous approaches to MOT in literature. Most MOT algorithms are based on Joint Probabilistic Data Association (JPDA) [2], Multi-Hypothesis Tracking [3], Probabilistic Multi Hypothesis Tracking [4] or Random

Fig. 1. MOT scenario with four cars (boxes) and four measurements (crosses).

Finite Sets (RFSs), where RFS based filters such as Probability Hypothesis Density filters [5], [6] and Poisson Multi-Bernoulli filters [7], [8] are well suited to handle object appearances and disappearances. An important example of RFS trackers is the Labeled Multi-Bernoulli (LMB) filter [9]. It considers all possible track-to-measurement associations as well as all possibilities of an object disappearing or persisting in the update step. The number of possible associations grows exponentially [9], which makes it computationally infeasible to exactly compute the posterior for growing numbers of objects or measurements.

So far, two approaches to approximate the LMB posterior are common. Firstly, Murty’s algorithm [10], which calculates the T best associations, and secondly Gibbs sampling, which generates associations based on their conditional likelihood [11]. Both algorithms are suitable approximations, but the Gibbs sampler clearly outperforms Murty’s algorithm regarding computation time, due to lower complexity [11]. Moreover, Gibbs sampling relies on the generation of random samples, thus producing nondeterministic results.

With MOT algorithms employed for safety critical functionalities such as navigation, nondeterminism can lead to problems regarding repeatability and validation [12]. Drawbacks of randomized approaches are, for example, that corner test cases might be hard to cover and that it is not sure whether a method will pass a test again, even if it has already passed it. Higher test capacities are required to calculate failure probabilities. Especially in the automotive area these concerns and the testability of systems pose a critical challenge due to the vast amount of scenarios in the real world [12]. Furthermore, due to the application of pseudonumber generators, randomized algorithms are not truly random and can have unforeseen side effects.

To avoid these issues, this paper focuses on deterministic methods to approximate the LMB posterior. Besides Murty’s algorithm, we will investigate Roecker’s algorithm [13], [14], a heuristic to generate joint associations for the JPDA filter.

This work was funded by the Lower Saxony Ministry of Science and Culture under grant number ZN3493 within the Lower Saxony “Vorab” of the Volkswagen Foundation and supported by the Center for Digital Innovations (ZDIN).

Laura M. Wolf and Marcus Baum are with the Data Fusion Group, Institute of Computer Science, University of Göttingen, Germany, laura.wolf@cs.uni-goettingen.de, marcus.baum@cs.uni-goettingen.de

Furthermore, we will propose a new method based on Herded Gibbs sampling [15], where a deterministic sampling method known as “herding” [16] is applied to Gibbs sampling. The deterministic Herded Gibbs has already proven to outperform randomized Gibbs sampling in image denoising and named entity recognition [15].

Our paper is structured as follows: Section II defines the problem, Section III presents the different association methods, followed by an evaluation on simulated data in Section IV.

II. PROBLEM FORMULATION

In MOT algorithms multiple hypotheses regarding which measurement originated from which object are considered. A joint association θ maps the set of objects to the set of measurements. As each object can generate at most one measurement and each measurement can only stem from one object, this is a one-to-one mapping. Since an object can disappear or be undetected even if it exists, the possibilities of object disappearance and missed detection are incorporated in the joint association. For this purpose, pseudo-measurements are used that are, however, not exclusive, since more than one object can disappear or be undetected in one time step. With $\{1, \dots, P\}$ the set of predicted objects and $\{1, \dots, M\}$ the set of measurements, a joint association $\theta : \{1, \dots, P\} \mapsto \{-1, 0, 1, \dots, M\}$ can be defined as:

$$\theta = (\theta_1, \dots, \theta_P), \quad (1)$$

where the individual association θ_i of object i has value -1 if object i does not exist, 0 if it exists, but is undetected, and value $j \in \{1, \dots, M\}$, if it is associated with measurement j . Measurements to which no object is assigned are considered to be clutter.

Due to the exponentially increasing number of possible joint associations, it could be advantageous to only consider likely associations. However, depending on noise and situation, unlikely associations might still be important, leading to a tradeoff between accuracy and complexity.

The likelihood of a joint association θ depends on the likelihood η_{i,θ_i} for an individual association θ_i . For the joint prediction and update scheme for the LMB filter [11], which estimates object disappearances and missed detections jointly with measurement associations, η is defined as:

$$\eta_{i,j} = \begin{cases} 1 - r_i p_S, & j = -1, \\ (1 - p_D) r_i p_S, & j = 0, \\ l_{i,j} r_i p_S, & j > 0, \end{cases} \quad (2)$$

for $1 \leq i \leq P$ and $-1 \leq j \leq M$, where r_i is the existence probability of object i , p_S is the survival probability, p_D the detection probability and $l_{i,j}$ the likelihood that measurement j was generated by object i . p_S and p_D have to be chosen depending on the specific setting and the likelihoods l depend on the applied measurement model.

Applying (2), the weight of association θ can be computed as [11]:

$$w^{(\theta)} = \prod_{i \in P} \eta_{i,\theta_i}. \quad (3)$$

Example 1: Consider the scenario in Fig. 1 with the following values for η :

η	-1	0	1	2	3	4
1	0.0	0.1	0.8	0.7	0.1	0.0
2	0.4	0.0	0.0	0.1	0.0	0.0
3	0.0	0.1	0.0	0.0	0.1	0.8
4	0.1	0.0	0.0	0.0	0.0	0.6

Two possible joint associations are $\theta = (1, -1, 4, -1)$ and $\theta' = (1, -1, 0, 4)$ with $w^{(\theta)} = 0.8 \cdot 0.4 \cdot 0.8 \cdot 0.1 = 0.0256$ and $w^{(\theta')} = 0.8 \cdot 0.4 \cdot 0.1 \cdot 0.6 = 0.0192$, showing that θ is more likely than θ' .

Given a set Θ of joint associations, the update step of an MOT filter can be computed considering only these instead of all possible associations. The LMB update, computing the updated existence probability r_i and spatial distribution $p_i(x)$ of object i , would then be given by the following:

$$r_i = \sum_{\theta \in \Theta, \theta_i \geq 0} \bar{w}^{(\theta)}, \quad (4)$$

$$p_i(x) = \frac{1}{r_i} \sum_{\theta \in \Theta, \theta_i \geq 0} \bar{w}^{(\theta)} p_i^{(\theta)}(x), \quad (5)$$

where $\bar{w}^{(\theta)}$ are the normalized weights of the joint associations and $p_i^{(\theta)}(x)$ is the posterior distribution of object i , given joint association θ . Note that for updating object i , only joint associations where i exists are considered.

Now, the complexity of the LMB update is determined by the size of Θ , which contains different associations depending on the algorithm applied.

III. DETERMINISTIC ASSOCIATION METHODS

In this section, three deterministic algorithms to compute joint associations will be discussed, namely Murty’s algorithm, Roecker’s algorithm and the novel approach based on Herded Gibbs sampling.

A. Murty’s Algorithm

Murty’s algorithm computes the T best solutions to the linear assignment problem, which can be solved optimally by the Hungarian method [17]. In order to apply Murty’s algorithm on the problem of finding the T best joint associations, η is converted to an extended cost matrix as described in [18]. It has the shape $P \times (2P + M)$ and consists of two square matrices, containing the likelihoods of object disappearance and missed detection on the diagonals, respectively. The off-diagonal entries are set to ∞ . The remainder of the cost matrix contains the likelihoods of the measurements being generated by the objects (shape $P \times M$). By taking the negative logarithm of all likelihoods, the problem of maximizing the LMB weights $w^{(\theta)}$, which are products of the likelihood η , see (3), is converted to the problem of minimizing a sum.

To compute the T best associations, Murty’s algorithm divides the set of possible associations into at most $P - 1$ disjoint subsets excluding the best association and computing the best association in each subset. Now the process is repeated with the next best association until T associations

are found. The Hungarian method can be implemented in $\mathcal{O}((2P + M)^3)$ [19], which yields a total complexity for Murty's algorithm of $\mathcal{O}(T \cdot (2P + M)^4)$ [18]. Other ranked assignment algorithms such as [20] are able to compute the T best associations even with cubic complexity.

Example 2: Considering the setting from Example 1, the three best joint associations are $\theta^1 = (1, -1, 4, -1)$, $\theta^2 = (2, -1, 4, -1)$ and $\theta^3 = (1, -1, 0, 4)$.

B. Roecker's Algorithm

Roecker proposed a heuristic to compute joint associations for a near optimal JPDA [13], [14]. The algorithm operates on a sorted list of individual associations L , in each step deleting one individual association θ_i (REMOVE_ONE, Algorithm 1). A joint association is generated using the remaining individual associations, this can be done in a greedy fashion, but it is also possible to compute the best association using the Hungarian method [17]. Note that the removed individual association can be considered again in the next step. This process can be repeated recursively, thus excluding pairs, triplets or higher tuples for each joint association (ROECKER and REMOVE_MULTI, Algorithm 1).

N_r is the number of individual associations to be removed for each joint association (e.g., individual associations, tuples, triples), N_a states the index, after which individual associations should be removed from L , N_b states the number of removed associations, after which the loop should break. We added this constant, since the individual association likelihoods in L decrease very fast and due to the greedy approach, the computed joint associations do not differ, if the excluded individual associations have low likelihood. Therefore, complexity can be reduced by ending the loops earlier without considerable changes in approximation quality.

Computing one greedy association has a complexity of $\mathcal{O}(P(M+2))$, thus, Roecker's algorithm has a complexity of $\mathcal{O}(N_b^{N_r} P(M+2))$ for removing all tuples up to length N_r .

Example 3: Consider again the setting from Example 1. At first, a greedy joint association is computed on all elements from L . The best individual associations are $\eta_{1,1} = 0.8$ and $\eta_{3,4} = 0.8$, which do not conflict with each other. $\eta_{1,2} = 0.7$ and $\eta_{4,4} = 0.6$ conflict with object 1 and measurement 4, respectively, which are both already associated. Continuing in this fashion yields the joint association $\theta^1 = (1, -1, 4, -1)$. Next, the best individual association $\eta_{1,1}$ is removed from L and another greedy joint association $\theta^2 = (2, -1, 4, -1)$ is computed. Then, $\eta_{1,1}$ is added to L again and the second best individual association $\eta_{3,4}$ is removed, which leads to $\theta^3 = (1, -1, 3, 4)$.

C. Herded Gibbs Sampling

Instead of computing joint associations θ , they can also be sampled. Since sampling from the joint distribution (3) is difficult, the Gibbs sampler samples individual associations θ_i from their conditional probabilities

$$P(\theta_i | \theta_1, \dots, \theta_{i-1}, \theta_{i+1}, \dots, \theta_P) = P(\theta_i | \bar{\theta}), \quad (6)$$

Algorithm 1 Roecker

function ROECKER(N_r, N_b, L)

$\Theta = \{\}$

Compute joint association θ of L , add to Θ

for $i = 1, \dots, N_r$ **do**

$\Theta = \text{REMOVE_MULTI}(0, i, N_b, L, \Theta)$

end for

return Θ

end function

function REMOVE_MULTI(N_a, N_r, N_b, L, Θ)

if $N_r = 1$ **then**

$\Theta = \text{REMOVE_ONE}(N_a, N_b, L, \Theta)$

else

for $j = N_a, \dots, N_a + N_b$ **do**

$\bar{L} = L$

Flag j^{th} individual association in $\bar{L}$ as removed

if $N_r > 2$ **then**

$\Theta = \text{REMOVE_MULTI}(j, N_r - 1, N_b, \bar{L}, \Theta)$

else

$\Theta = \text{REMOVE_ONE}(j, N_b, \bar{L}, \Theta)$

end if

end for

end if

return Θ

end function

function REMOVE_ONE(N_a, N_b, L, Θ)

for $i = N_a + 1, \dots, N_a + 1 + N_b$ **do**

$\bar{L} = L$

Flag i^{th} individual association in $\bar{L}$ as removed

Compute joint association θ of $\bar{L}$

Add θ to Θ , if it is not a duplicate

end for

return Θ

end function

with $\bar{\theta} := [\theta_1^{(t)}, \dots, \theta_{i-1}^{(t)}, \theta_{i+1}^{(t-1)}, \dots, \theta_P^{(t-1)}]$. Gibbs sampling has been successfully applied to decrease the complexity of the update steps of the LMB and Generalized Labeled Multi-Bernoulli filters, but it is randomized. The work [15] applied "Herding" to Gibbs sampling, thus creating a deterministic version of it. Herding [16] is a procedure which can be applied to compute a sequence of pseudo-samples matching a normalized discrete probability distribution $\mathbf{p}$. The t^{th} sample $\mathbf{x}^{(t)} \in \mathcal{X} \subseteq \mathbb{R}^n$ is computed as:

$$\mathbf{x}^{(t)} = \arg \max_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{w}^{(t-1)}, \phi(\mathbf{x}) \rangle, \quad (7)$$

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} + \mathbf{p} - \phi(\mathbf{x}^{(t)}), \quad (8)$$

where the weight vector $\mathbf{w}$ has the same dimension as $\mathbf{p}$ and the feature map ϕ is set to be the indicator vector, which consists of zeros and a single one at the position indicated by $\mathbf{x}$ [15], $\langle \cdot, \cdot \rangle$ denotes the inner product.

In [15] herding is applied to Gibbs sampling [21] by generating deterministic samples from the conditional probability

distributions. Herded Gibbs sampling requires to maintain a weight vector for each conditional distribution. Since there are P objects to sample associations for and for each of the other objects exist $M + 2$ association possibilities, there are $P(M + 2)^{(P-1)}$ conditional distributions to sample from. This resembles the worst case exponential space complexity of Herded Gibbs sampling. But as one object can only be assigned to one measurement, this number reduces drastically. Furthermore, the number of conditional distributions for which weight vectors have to be stored is limited by the number of samples that are actually drawn.

Algorithm 2 Herded Gibbs sampling

```

function HERDED_GIBBS( $\eta, T$ )
  Initialize  $\theta$  with zeros,  $\Theta = \{\}$ 
  for  $t = 1, \dots, T$  do
    for  $i = 0, \dots, P$  do
      Lookup  $\mathbf{w}^{i, \bar{\theta}}$  in hash table
      if  $\mathbf{w}^{i, \bar{\theta}}$  is not in hash table then
         $\mathbf{w}^{i, \bar{\theta}} = P(\theta_i | \bar{\theta}) / \sum_j P(\theta_i = j | \bar{\theta})$ 
      end if
       $\theta_i = \arg \max \mathbf{w}^{i, \bar{\theta}}$ 
       $\mathbf{w}^{i, \bar{\theta}} = \mathbf{w}^{i, \bar{\theta}} + P(\theta_i | \bar{\theta}) / \sum_j P(\theta_i = j | \bar{\theta})$ 
       $\mathbf{w}_{\theta_i}^{i, \bar{\theta}} = \mathbf{w}_{\theta_i}^{i, \bar{\theta}} - 1$ 
      Save  $\mathbf{w}^{i, \bar{\theta}}$  to hash table
    end for
    Add  $\theta$  to  $\Theta$ , if it is not a duplicate
  end for
  return  $\Theta$ 
end function

```

The conditional probabilities $P(\theta_i | \bar{\theta})$ of the joint distribution (3) are given by [18]:

$$P(\theta_i = j | \bar{\theta}) \propto \begin{cases} 0, & \text{if } j > 0 \text{ and } j \text{ is assigned,} \\ \eta_{i,j}, & \text{else.} \end{cases} \quad (9)$$

The probability that object i is assigned to measurement j is zero, if j is already associated with another object. Values -1 and 0 are excluded since they resemble object disappearance and missed detection, to which multiple objects can be assigned.

Looking up the weight vector $\mathbf{w}^{i, \bar{\theta}}$ (Algorithm 2) can be implemented in constant time using a Hash Table. If keeping track of assigned measurements, we can compute the conditional probabilities $P(\theta_i | \bar{\theta})$ in $\mathcal{O}(M + 2)$, thus yielding a total complexity of $\mathcal{O}(TP(M + 2))$ for computing T joint associations. In contrast to Herded Gibbs sampling, Randomized Gibbs sampling does not maximize a weight vector, but samples directly from the conditional distribution $P(\theta_i | \bar{\theta})$, which can also be done in linear time. This results in the same complexity for both Gibbs samplers.

Example 4: Consider again the setting from Example 1. θ is initialized with all objects being undetected. We have $\mathbf{w}^{1, \bar{\theta}} = \eta_{1, \cdot} / \sum_j \eta_{1, \cdot} = [0.0, 0.06, 0.47, 0.41, 0.06, 0.0]$, since no measurements have been associated yet. Maximizing yields $\theta_1 = 1$. Updating the weight vector yields $\mathbf{w}^{1, \bar{\theta}} =$

$[0.0, 0.12, -0.06, 0.82, 0.12, 0.0]$. Continuing this way yields $\theta^1 = \theta^2 = (1, -1, 4, -1)$ and $\theta^3 = (2, -1, 4, -1)$. Once the weight vectors balance out due to reusing, the samples start to differ more.

IV. EVALUATION

In order to compare the different data association methods, we combined them with the LMB filter, which was implemented¹ in Python, adapted from the MATLAB implementation² of [22] with Gaussian mixture states and a constant velocity model. It was combined with the joint prediction and update described in [11]. We used fixed birth locations for simplicity, although an adaptive birth strategy would also be possible. Scenario A consists of two objects closing in on each other, moving in parallel and then moving apart. Scenario B is defined as in [23], [22] and includes crossing and appearing/disappearing objects as well as objects from the same origin. Each association method was evaluated on 100 Monte Carlo runs of the filter. In order to compare the approximation accuracy of the different data association methods, the mean OSPA [24] metric with parameters $p = 1$ and $c = 100$ over all runs as well as the mean number of estimated objects over all runs were evaluated.

We evaluated Murty's algorithm with 100 joint associations, Herded Gibbs sampling as well as randomized Gibbs sampling with 1000 joint associations and Roecker's algorithm with $N_r = 2, 3$, $N_b = 10$ and greedy association computation. Using the Hungarian method instead of the greedy approach did not improve the results at the cost of a higher computation time and is therefore omitted here.

In order to compare the convergence behaviour of both Gibbs sampling methods, we evaluated the OSPA metric averaged over 100 Monte Carlo runs and all time steps of Herded Gibbs sampling and randomized Gibbs sampling on scenario B with a varying number of samples, $T \in \{10, 20, \dots, 100, 200, \dots, 1000\}$.

In both scenarios, Murty's algorithm and both Gibbs samplers clearly outperform Roecker's algorithm regarding the OSPA metric (Fig. 2a and 3a). Both tested variants of Roecker's algorithm estimate too many tracks in scenario A (Fig. 2b) and have difficulties with disappearing objects (Fig. 3b). An example run of Roecker2 on scenario A can be seen in Fig. 4a. Increasing the length of removed tuples and therefore the number of generated associations, does not necessarily yield better results, since Roecker2 performed better on scenario A (Fig. 2a and 2b) and in scenario B before time step 20 (Fig. 3a and 3b).

There are almost no visible differences between Murty's algorithm, randomized Gibbs sampling and Herded Gibbs sampling in both scenarios. All three of them show a very low OSPA error (Fig. 2a and 3a). Furthermore, the number of false tracks is very small in both scenarios (Fig. 2b and 3b). An example trace of Herded Gibbs sampling is given in Fig. 4b.

¹<https://github.com/Fusion-Goettingen>

²<http://ba-tuong.vo-au.com/codes.html>

Fig. 2. Comparison of different data association methods on scenario A. Mean OSPA metric (a) and mean estimated cardinality (b) of 100 Monte Carlo runs of the LMB filter with different data association methods.

Fig. 3. Comparison of different data association methods on scenario B. Mean OSPA metric (a) and mean estimated cardinality (b) of 100 Monte Carlo runs of the LMB filter with different data association methods.

Fig. 4. Exemplary single runs of LMB filter with Roecker2 greedy (a) and Herded Gibbs (b) on scenario A and B, respectively. Clutter and object generated measurements of all time steps are shown at the same time. Estimated object trajectories are shown as colored lines.

Fig. 5. Comparison of both Gibbs sampling variants with varying numbers of samples T . OSPA averaged over all time steps and 100 Monte Carlo runs of the LMB filter on scenario B.

The OSPA error of Herded Gibbs sampling converges with a smaller number of samples than randomized Gibbs sampling (Fig. 5). Although the difference is very small, it also shows a slightly lower OSPA error in general. Therefore, Herded Gibbs sampling might be applied with a smaller number of samples than randomized Gibbs sampling, decreasing computation time. With our Python implementation Herded Gibbs sampling took 1.7 to 2.2 times the computation time of randomized Gibbs sampling, while the ratio decreased with increasing sampling size T .

V. CONCLUSION

The data association problem in MOT is a highly complex task with no easy solution. Conceptually simple heuristics such as the greedy version of Roecker's algorithm can generate useful results, although far better results can be achieved by applying one of the other algorithms. Furthermore, the parameterization of Roecker's algorithm seems to be of importance, since Roecker2 is the better choice in scenario A and Roecker3 in scenario B. It could be investigated whether tuning N_r and N_b improves the results, even though this requires extra effort and knowledge of the scenario. Moreover, the theoretical complexity is similar to the complexity of Herded Gibbs sampling, which clearly outperforms both variants of Roecker's Algorithm.

Herded Gibbs sampling achieves the same approximation quality as Murty's algorithm and randomized Gibbs sampling while having the same theoretical complexity as randomized Gibbs sampling and furthermore being deterministic. The comparison of both Gibbs samplers with varying sampling sizes even suggests, that Herded Gibbs sampling could be applied with a smaller sampling size while yielding the same approximation quality, which compensates for the higher computation time. Therefore, Herded Gibbs sampling is a suitable deterministic alternative to randomized Gibbs sampling in data association for MOT.

Future work will include further optimization of the computation time of Herded Gibbs sampling and investigating whether it can be applied to other data association problems.

REFERENCES

- [1] J. Vermaak, S. Godsill, and P. Perez, "Monte carlo filtering for multi-target tracking and data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 1, pp. 309–332, Jan. 2005.
- [2] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, Jul. 1983.
- [3] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
- [4] R. L. Streit and T. E. Luginbuhl, "Probabilistic multi-hypothesis tracking," Naval Underwater Systems Center Newport RI, Tech. Rep. NUWC-NPT-TR-10428, Feb. 1995.
- [5] R. Mahler, "Multitarget Bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.
- [6] Á. F. García-Fernández and L. Svensson, "Trajectory PHD and CPHD filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 22, pp. 5702–5714, Nov. 2019.
- [7] K. Granström, L. Svensson, Y. Xia, J. Williams, and Á. F. García-Fernández, "Poisson multi-Bernoulli mixtures for sets of trajectories," *arXiv:1912.08718 [cs, eess, stat]*, Dec. 2019.
- [8] Á. F. García-Fernández, L. Svensson, J. L. Williams, Y. Xia, and K. Granström, "Trajectory poisson multi-Bernoulli filters," *arXiv:2003.12767 [cs, stat]*, Jul. 2020.
- [9] S. Reuter, "Multi-object tracking using random finite sets," Dissertation, Universität Ulm, Sep. 2014.
- [10] K. G. Murty, "Letter to the editor—An algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, no. 3, pp. 682–687, Jun. 1968.
- [11] S. Reuter, A. Danzer, M. Stübler, A. Scheel, and K. Granström, "A fast implementation of the labeled multi-Bernoulli filter using Gibbs sampling," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 765–772.
- [12] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.
- [13] J. A. Roecker, "A class of near optimal JPDA algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 504–510, Apr. 1994.
- [14] K. Romeo, D. F. Crouse, Y. Bar-Shalom, and P. Willett, "The JPDAF in practical systems: Approximations," in *Signal and Data Processing of Small Targets 2010*, vol. 7698. International Society for Optics and Photonics, Apr. 2010, p. 769811.
- [15] Y. Chen, L. Bornn, N. De Freitas, M. Ekelin, J. Fang, and M. Welling, "Herded Gibbs sampling," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 263–291, Jan. 2016.
- [16] M. Welling, "Herding dynamical weights to learn," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. Montreal, Quebec, Canada: Association for Computing Machinery, Jun. 2009, pp. 1121–1128.
- [17] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.
- [18] B. Vo, B. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 1975–1987, Apr. 2017.
- [19] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171–175, Oct. 1986.
- [20] C. R. Pedersen, L. Relund Nielsen, and K. A. Andersen, "An algorithm for ranking assignments using reoptimization," *Computers & Operations Research*, vol. 35, no. 11, pp. 3714–3726, Nov. 2008.
- [21] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
- [22] S. Reuter, B. Vo, B. Vo, and K. Dietmayer, "The labeled multi-Bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, Jun. 2014.
- [23] Ba-Tuong Vo, Ba-Ngu Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 409–423, Feb. 2009.
- [24] D. Schuhmacher, B. Vo, and B. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.